

OSSERVATORIO VESUVIANO



RETE MOBILE

**STAZIONI SISMICHE DIGITALI LENNARTZ MARS-LITE
PREPARAZIONE E DECODIFICA**

a cura di M.Castellano

Rel. 1.0

Novembre 1999

STAZIONI SISMICHE DIGITALI LENNARTZ MARS-LITE PREPARAZIONE E DECODIFICA

a cura di M.Castellano
rel. 1.0 (11.99)

Le stazioni sismiche digitali Lennartz MarsLite a 3 componenti in dotazione alla Rete Mobile dell'Osservatorio Vesuviano sono l'ultima evoluzione della strumentazione sismica della ditta tedesca e prevedono per l'acquisizione dei dati un drive magneto-ottico Fujitsu con interfaccia SCSI-2 da 540Mb, utilizzabile con dischi magneto-ottici standard da 3.5".

Questo manuale non vuole in nessun modo sostituirsi al manuale della Lennartz, a cui si rimanda per una completa trattazione su funzioni e programmazione, nè essere un manuale sulle procedure di installazione, ma vuole dare alcune informazioni indispensabili per la preparazione dei dischi magneto-ottici e la decodifica dei dati registrati.

La Lennartz fornisce una serie di programmi DOS con i quali è possibile preparare i dischi ed estrarre da essi i dati. L'elenco dei programmi disponibili e i relativi file *readme* con le informazioni sul loro uso è allegato a questo breve manuale in cui, comunque, sarà riportata una breve descrizione delle funzioni dei programmi più utili.

Non viene fornito alcun programma per l'analisi dei dati, per cui è necessario disporre di programmi in grado di trattare file ASCII o file SUDS.

La preparazione dei dischi e la decodifica dei dati acquisiti con le stazioni digitali LENNARTZ MARS-LITE può essere effettuata da qualsiasi calcolatore dotato di unità magneto-ottica SCSI da 540 o 640Mb.

Nell'attuale configurazione del Laboratorio Rete Mobile presso la sede dell'Osservatorio Vesuviano di Via Diocleziano 328 è presente un calcolatore espressamente dedicato al trattamento dei dati MARS-LITE:

Calcolatore *Lite* PII-400 SCSI (PW xxxx; WIN95)

Disco C - 9.1Gb

Disco D - Unità magneto-ottica Fujitsu DynaMo esterna 640Mb (da accendere prima del calcolatore)

Disco E - CD-ROM 32X

Di seguito vengono descritte le procedure necessarie alla preparazione dei dischi magneto-ottici ed alla decodifica dei dati registrati. Per ogni procedura vengono riportati i programmi da utilizzare con le relative funzioni.

N.B. Per l'esecuzione dei programmi Lennartz è necessario riavviare il computer in modalità DOS.

Nel disco C è stata creata la directory **MARS** in cui sono presenti tutti i programmi.
L'esecuzione di ogni programma prevede la seguente sintassi:

nomeprogramma -s 2

dove 2 individua il numero di ID dell'unità SCSI attualmente installata.

PREPARAZIONE DEI DISCHI

Prima dell'impiego sulle stazioni MarsLite i dischi magneto-ottici vanno formattati utilizzando esclusivamente il programma specifico **MOFORMAT**. La sintassi sarà quindi:

C:\mars\moformat -s 2

Alla successiva richiesta di conferma digitare [y]. La formattazione di un disco da 540Mb richiede circa 15 minuti ed il processo può essere seguito dall'avanzamento di un cursore. Quando la linea sarà completata la formattazione sarà conclusa ed il programma tornerà al prompt.

A questo punto bisogna creare delle partizioni sul disco usando il programma **PARTEDIT**. Sintassi:

C:\mars\partedit -s 2

E' necessario creare 4 partizioni:

DATA	spazio destinato alla registrazione dei dati
MONITOR	spazio destinato alla registrazione in continuo del segnale del canale 0
LOGGING	spazio destinato alla registrazione delle informazioni relative a tutte le operazioni effettuate dalla stazione.
SETUP	spazio destinato alla registrazione di un file di setup

Dal menù di PARTEDIT è possibile effettuare una partizione automatica →[a] che definisce il numero di blocchi, e quindi lo spazio, riservato ad ogni partizione. E' possibile modificare l'impostazione automatica in funzione delle specifiche necessità richiamando le singole partizioni [m(onitor), d(ata), l(ogging), s(etup)]. Per esempio è possibile azzerare lo spazio per MONITOR, nel caso non si voglia disporre di tale funzione (assegnazione blocchi per **MONITOR = 0**) ed assegnare a DATA tutto lo spazio libero (assegnazione blocchi per **DATA = -1**).

La partizione MONITOR è utile per avere un monitoraggio continuo del segnale registrato al fine di individuare gli eventi avvenuti durante il periodo di funzionamento.

Dopo aver definito la partizione bisogna scriverla sul disco con il comando **w(rite)**. E' possibile leggere le partizioni impostate su un disco con il comando **r(ead)**.

Dopo tali operazioni il disco è pronto per essere inserito nella stazione MarsLite per l'acquisizione.

DECODIFICA

La decodifica viene effettuata riavviando il computer in modalità DOS ed utilizzando una serie di programmi proprietari forniti dalla Lennartz con i quali è possibile, tra l'altro, estrarre i dati in due formati: ASCII e/o SUDS.

Altri programmi consentono di controllare il funzionamento della strumentazione, il contenuto del disco, lo spazio utilizzato ed altre funzioni. Di seguito sono elencati i programmi più utilizzati per la decodifica dei dati con le loro funzioni, rimandando al contenuto dell'appendice per i dettagli su tutto il software disponibile.

LOGINFO - consente di visualizzare la lista (data, ora e tipo) delle operazioni contenute nella partizione **logging** (mount ed unmount del disco, stato della batteria, sincronismo). Si consiglia di scaricare il contenuto della partizione in un file di testo per avere sempre sotto controllo la “storia” del disco registrato → **loginfo -s 2>nomefile**

EVLIST - legge le finestre di registrazione contenute nel disco definendo il tempo di inizio e di fine di ogni finestra. Crea due file batch che consentono l'estrazione dei dati in formato ASCII e SUDS rispettivamente (**ASCII.BAT** e **DUMP_ALL.BAT**) per ogni finestra registrata.

LT2ASCII - consente l'estrazione dei dati in formato ASCII. Usa la sintassi:

lt2ascii -s 2 -f iniziofinestra(AAMMGHHmmss) -l finefinestra(AAMMGHHmmss)

LT2SUDS - consente l'estrazione dei dati in formato SUDS. Usa la sintassi:

lt2suds -s 2 -f iniziofinestra(AAMMGHHmmss) -l finefinestra(AAMMGHHmmss)

sudplot /p3 /m1000 event.sud

rename event.sud MMGGHHmm.ssS

MONREAD - legge il contenuto della partizione MONITOR e crea un file visualizzabile con programmi tipo SUDSDRUM (IASPEI)

Le procedure di decodifica cambiano leggermente a secondo che l'acquisizione sia stata effettuata a trigger o in continuo.

ACQUISIZIONE A TRIGGER

Inserire il disco magneto-ottico nell'unità D. Da **C:\>** creare una directory col nome della stazione ed il numero progressivo del disco di registrazione (es. **ABC123**).

Da questa directory lanciare il programma **evlist** con la sintassi corretta [**C:\ABC123\evlist -s 2**]. Alla fine del processo saranno creati due file batch che consentono l'estrazione dei dati in formato ASCII e SUDS rispettivamente (**ASCII.BAT** e **DUMP_ALL.BAT**) per ogni finestra registrata.

ASCII.BAT

Per ogni file-evento vengono creati tre file (uno per ogni canale) con l'etichetta:

MMGGHHmm.c00 componente verticale

MMGGHHmm.c01 componente longitudinale

MMGGHHmm.c02 componente trasversale

Le informazioni relative al file (sigla stazione, frequenza di campionamento, tempo del primo campione) sono riportate nella prima linea dello stesso file.

DUMP_ALL.BAT

Per ogni file-evento viene creato un unico file con l'etichetta:

MMGGHHmm.ssS

Una volta estratti, entrambi i formati possono essere trattati con il programma **PITSA**.

ACQUISIZIONE IN CONTINUO

Inserire il disco magneto-ottico nell'unità D. Da **C:\>** creare una directory col nome della stazione ed il numero progressivo del disco di registrazione (es. **ABC123**).

In questo caso non è necessario utilizzare il programma **evlist**.

Una volta saputo (dal MONITOR o da informazioni di reti fisse monitorate) l'elenco degli eventi presenti sul disco di cui si vuole effettuare l'estrazione, bisogna editare i file ***.BAT** contenenti le informazioni previste dai file standard ASCII.BAT o DUMP_ALL.BAT.

ASCII.BAT

```
lt2ascii -s 2 -f AAMMGHHmmss -l AAMMGHHmmss  
                (inizio)                (fine)
```

DUMP_ALL.BAT

```
lt2suds -s 2 -f AAMMGHHmmss -l AAMMGHHmmss  
sudplot /p3 /m1000 event.sud  
rename event.sud MMGGHHmm.ssS
```

In questo modo è possibile impostare finestre temporali delle dimensioni volute; bisogna comunque tenere in considerazione le dimensioni massime consentite dal programma di trattamento.

I SEGUENTI PROGRAMMI ESEGUIBILI SONO PRESENTI NELLA DIRECTORY MARS:

BLKREAD.EXE Reads a single block of data from the disk, displaying its contents in hexadecimal and ASCII. If the block contains a recognizable MARSlite header structure, also displays the header contents.

CHK_ASPI.EXE Checks the availability of the ASPI software interface, probes all SCSI addresses (0...7) for connected devices.

DATAINFO.EXE Displays header information for every data block found on the disk.

EVLIST.EXE Creates an "event list" from the data partition, trying to recognize clusters of chronologically consecutive blocks as "events". Also creates a batch file (DUMP_ALL.BAT) which calls lt2suds.exe for each event found, and renames the event. File name is MMDDHHmm.ssS where MM is month (01..12), DD is day (01..31), HH is hour (00..23), mm is minute (00..59), and ss is seconds (00..59). The final 'S' is the letter S. Example: 10042316.04S means Oct 04, 23:16:04.

LOGINFO.EXE Displays the contents of the logging partition, one log entry per line.

LT2SUDS.EXE Converts a user-defined time window (-f, -l) to a PC-SUDS event. This program suffers from the inherent "real mode" limitation of 64 kB maximum size for a data structure. Long time series can not be created. You will have to split them up. Output is in microvolts (floating point format). The output file name is EVENT.SUD. If you wish to keep this file, rename it to a different name. You can use PITSA to view this file. By changing a compilation constant you can create integer output in nanovolts. The version of SUDSPLOT that I have been using during development has a bug which does not allow floating point data to be plotted, but maybe that has been solved in the mean time. In this case you can use SUDSPLOT as an alternative to PITSA.

LT2ASCII.EXE Converts a user-defined time window (-f, -l) to an ASCII file (one file per channel will be created). Unlike lt2suds.exe, this program can also deal with very large time series. The first line of each ASCII file contains a comment describing the station name, sampling interval, and time; subsequent lines contain one value expressed in microvolts.

MOFORMAT.EXE "Formats" an MO disk by overwriting every block on the disk. THIS IS NECESSARY if you want to re-use a previously recorded disk. Simply overwriting the partition table will NOT make it re-useable!!!

MONINFO.EXE Displays header information for every monitor block found on the disk.

MONREAD.EXE Reads monitor data from the disk, creating one or more PC-SUDS data files that can be viewed using SUDSDRUM. A new file is started for each day. File name is MNyymmdd.SUD where yy is the year, mm is the month, and dd is the day (e.g. MN950314.SUD for the monitor data of March 14, 1995). Output is in NANOVOLTS.

PARTEDIT.EXE This is a partition editor which allows you to read, modify, and write a MARSlite partition table structure to and from an MO disk.

PARTINFO.EXE This program just reads a partition table off the disk and displays information (where the partition is located and how much it is occupied).

WHAT.EXE A small auxiliary program for displaying version information in a .EXE program. Usage: what filename

WR_SETUP.EXE Copies a user-defined, existing file into the SETUP partition of a MARSlite disk.

Program name:

BLKREAD.EXE

Description:

This program reads a single block (512 bytes) from a MARSlite disk (or from any other MO disk, for that matter). The contents of the block are displayed in hexadecimal and ASCII notation. A non-printable byte is replaced by a period ('.').

If the block contains a recognizable MARSlite block header, an excerpt from the block header is printed after the whole block has been displayed.

You will see the starting date and time of this block, the station ID code, the sampling interval in msec, and the maximum amplitude (in microvolts) found in this block.

Here is an example:

Block 34:

```
0x0000: 6c 65 03 05 54 45 53 54 00 00 00 00 3c d7 66 2f le..TEST....<.f/
0x0010: 02 03 24 01 05 12 06 b7 7e 0a b6 f6 ee 01 6e 43 ..$.~.....nC
... many lines omitted ...
0x01d0: 0d 20 0e 76 9e ea 0d d8 05 e0 2e 1c f6 7d 66 44 . .v.....}fD
0x01e0: 4e ef 0e b4 fe aa ee 08 a6 50 9e 58 7e 1a 96 bb N.....P.X~...
0x01f0: 2e ac 06 f0 be 3b 56 2a ae 05 36 da 8e eb de 24 .....;V*..6....$
```

ID "TEST" 1995-03-15 12:02:04 ch2 sc 32 max 9344 microvolts

After starting the program, and after each block displayed, a prompt line asks you to enter a block number. If you enter a negative block number this will terminate the program.

You can enter a "+" or a "-" to indicate the next or previous block, respectively. Alternatively, simply give a block number.

Calling convention:

BLKREAD -s <SCSI_address>

Documentation modification date:

95/03/16

Program name:

CHK_ASPI

Description:

This program checks the availability of ASPI services. If an ASPI compliant driver is available, this program shows the name and version of the ASPI driver and SCSI host interface, and informations about the connected device(s). All SCSI target addresses (0...7) are probed. Targets implementing LUNs (Logical Unit Numbers) other than 0 are not supported.

Calling convention:

chk_aspi

Documentation modification date:

95-03-15

Program name:

DATAINFO.EXE

Description:

This program lists the contents of the DATA partition on a MARS Lite disk. For each block, one line of header information is printed. For data blocks, in addition to the time, station name, channel number, sampling interval, duration of the block, and maximum amplitude, also the trigger index is printed (i.e. the sample number where the user-defined STA/LTA trigger criterium was fulfilled for the first time within the block). For blocks that do not contain a trigger, nothing will be printed here. Here is an example line:

```
178 DAT DGG c00 smp 16 sc 32 dur 8 max 840 1995-03-20 16:21:12 tr 276
+++ sampling interval (milliseconds)
+++ scale (uV per count)
+++ duration of blocks (seconds)
+++++ max. amplitude (counts)
date and time of start of block ++++++
trigger index (where trigger criterium was fulfilled first time) +++
```

Calling convention:

DATAINFO -s <scsi_address>

Documentation modification date:

95/03/30

Program name:

EVLIST.EXE

Description:

This program generates an 'event list' file from the contents of the DATA partition on a MARSlite disk.

An event is defined as a sequence of contiguous blocks. As soon as a "time tear" occurs, an event will be declared finished.

For each event found, this program writes a line containing the begin and end times of the event.

In addition, a file called "DUMP_ALL.BAT" will be written in parallel. For each event found, one command line each for the LT2SUDS program (which is also part of this distribution) and for the SUDSPLOT program are created.

The SUDSPLOT command line will cause the event file to be read, and then be plotted and output to the hardcopy device (which depends on how you have set up the SUDSUTIL.INI file). Finally, the event created by LT2SUDS (which is always called 'EVENT.SUD') is renamed. The new file name is MMDDhhmm.ss

(MM: month, DD: day, hh: hour, mm: minute, ss: second).

You can of course edit the DUMP_ALL.BAT file to suit your particular needs.

In addition to the file DUMP_ALL.BAT, newer releases of EVLIST.EXE also create a file ASCII.BAT which contains similar commands for LT2ASCII.EXE (conversion to ASCII format).

Calling convention:

EVLIST -s <scsi_address>

Documentation modification date:

96/01/16

Program name:

LOGINFO.EXE

Description:

This program prints the contents of the LOGGING partition on a MARSlite disk.

Each entry is printed on one line.

Calling convention:

LOGINFO -s <SCSI_address>

Documentation modification date:
95/03/23

Program name:

LT2SUDS.EXE

Description:

This program converts a certain time segment (normally defined by using the '-f' and '-l' command line options) from MARSlite internal format to the PC-SUDS 1.x format.

Depending on which symbols are defined at compile time, LT2SUDS either outputs floating point numbers (microvolts), or raw digital counts (the latter is much more difficult to handle because you will need to take the scale into account manually, and you need to know that there is an implied "binary point" at bit 14). For using the raw digital counts format, you need to multiply by "scale" and then divide by 16384 in order to arrive at microvolts.

Since LT2SUDS was compiled with a 16-bit compiler it suffers from the inherent limitations of DOS and real mode, i.e. the maximum size of any data item (e.g. an array) is 64 kB. This limits the size of time series that can be created.

Unfortunately, in order to be useable by SUDSPLOT, a SUDS trace must have been created with a single 'sudswrite()' command. Unlike MONREAD, LT2SUDS can not simply "append" blocks. When LT2SUDS finds it would need to allocate more than 64 kB of data per trace, it will artificially terminate the event at this point, and write out whatever fits in the 64 kB available.

The output event is always called 'EVENT.SUD'. If you want a more meaningful name, edit and compile

Calling convention:

LT2SUDS -s <SCSI_address> [-f YYMMDDhhmmss] [-l YYMMDDhhmmss]

Documentation modification date:
95/04/04

Program name:

LT2ASCII.EXE

Description:

This program converts a certain time segment (normally defined by using the '-f' and '-l' command line options) from MARSlite internal format to ASCII format.

LT2ASCII stops at the time indicated by '-l', or at a "break of event", i.e. if the next data block does not match the time of the block previously read.

Output files are named "MMDDHHmmSS.cXX" where MM= month, DD = day, HH = hour, mm = minutes, SS = seconds, XX = channel number. Channels are numbered 00 through 02.

Example:

0116122304.c01

Jan 16, 12:23:04, channel 1.

The first line of the ASCII file is a comment line containing the station name, sampling interval in milliseconds, and the exact time of the first sample.

Calling convention:

LT2ASCII -s <SCSI_address> [-f YYMMDDhhmmss] [-l YYMMDDhhmmss]

Documentation modification date:

96/01/16

Program name:

MAXAMP

Description:

This program generates a list of maximum amplitudes found in data blocks.

As it is compiled in the distribution it only looks at channel 0; to change this you will need to modify the source.

The largest maximum amplitude found so far is marked by a trailing asterisk.

Calling convention:

MAXAMP -s <SCSI_address> [-f YYMMDDhhmmss] [-l YYMMDDhhmmss]]

Documentation modification date:
95/10/30

Program name:

MOFORMAT

Description:

This program "formats" a MARSlite MO disk. It does not really perform a low-level formatting operation. Rather, it overwrites all blocks on the disk with the same pattern (a 64-byte string containing the date and time of formatting).

Since this program will irrevocably destroy the contents of an MO disk it asks for your confirmation before it starts to write.

During "formatting", a block counter is updated so that you can see how far you are.

Depending on the disk capacity, typical formatting times are between 15 and 30 minutes.

After having "formatted" the last block, the program will test reading a few blocks from the disk, and then terminate.

Calling convention:

MOFORMAT -s <SCSI_address>

Documentation modification date:
95-03-15

Program name:

MONINFO.EXE

Description:

This program lists the contents of the MONITOR partition on a MARSlite disk. For each block, one line of header information is printed. Here is an example line:

```
98713 MON DGG c00 smp 16 sc 32 dur 400 max 3358 1995-03-20 16:15:44
++++++ absolute block number on the disk
+++ MON: monitor block, DAT: data block
++++ four-character station ID (as set by the user on MARSlite)
+++ channel number (always 0 for monitor blocks)
```

Calling convention:

MONINFO -s <scsi_address>

Documentation modification date:
95/03/30

Program name:

PARTEDIT

Description:

This program is a partition table editor which allows you to read, modify, and write the partition table on a MARSlite MO disk.

After startup, the program prompts you for single-character input. Type 'h' to get a brief help screen.

When you choose the A(utomatic setting, a "standard" partition table will be created in memory. This partition table allocates 2 blocks (1 kB) for the SETUP partition, 2048 blocks (1 MB) for the LOGGING partition, 20% of the remaining capacity for the MONITOR partition, and the whole rest for the DATA partition.

Calling convention:

PARTEDIT -s <SCSI_address>

Documentation modification date:
95-03-15

Program name:

PARTINFO

Description:

This program prints information about the partition table on a MARSlite MO disk.

Besides reading the partition table sizes from the MO, this program also tries to figure out how much space is used in each of the four partitions. For the DATA and MONITOR partition, the number of used blocks is given. For the LOGGING partition, the program counts the number of logging entries (each entry is 64 bytes in size).

For the SETUP partition, there are only two possible states: EMPTY and Occupied. "Occupied" means that there is a block in that partition which contains the correct 'magic number', i.e. '#S'.

Calling convention:

PARTINFO -s <SCSI_address>

Documentation modification date:
95-03-29

Program name:

WR_SETUP.EXE

Description:

This program copies an existing DOS file (whose name is asked from the user) to the SETUP partition on a MARSlite disk.

When MARSlite tries to load the contents of the SETUP partition it expects the first two bytes to be '#' and 'S'. The WR_SETUP program prepares its output buffer accordingly; the first line reads

#Setup file created by WR_SETUP.EXE

Hence you will not need to include the "#S" string at the beginning of your setup file (it will not do any harm if it is there anyway).

You should have prepared an ASCII text file containing MARSlite commands before calling up WR_SETUP. The file should contain one command per line. It may also contain comments (using the '#' character as a comment lead-in).

Since MARSlite uses the Unix convention of line termination (i.e. a single line feed character (hex 10) terminates a line), and DOS uses carriage return AND line feed (hex 13 and hex 10), carriage return characters are 'translated' to line feed characters, turning each DOS line termination into TWO line terminations.

The WR_SETUP program does NOT perform any plausibility checks. The syntax of the statements in your input file will be checked only at the time of reading the contents of the SETUP partition into MARSlite.

In order to know when to stop interpreting, MARSlite requires either an "exit" statement, or an end-of-file character (hex 4, CTRL-D). Again, WR_SETUP is smart enough to pad your input file with CTRL-D characters. Even if you forget to include the "exit" statement at the end, no problem.

Calling convention:

WR_SETUP -s <SCSI_address>

Documentation modification date:
95/03/28

Program name:

WHAT

Description:

This is a clone of the Unix what(1) command. It reads a file (binary or text) and scans it for the "magic string" @(#) which is used by SCCS (Source Code Control System) to indicate a version number.

Calling convention:

WHAT [-s] filename

-s: stop after first occurrence of "magic pattern"

Example:

WHAT CHK_ASPI.EXE

Documentation modification date:

95-03-15